

A Phenomenological Approach to Integrating Gaussian Beam Properties and Speckle into a Physically-Based Renderer

S. Bergmann¹, M. Mohammadikaji¹, S. Irgenfried¹, H. Wörn¹, J. Beyerer^{1,2}, C. Dachsbacher¹

¹Karlsruhe Institute of Technology (KIT), Germany

²Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany

Abstract

Coherent light exhibits properties that cannot be modeled by geometrical optics. Those properties include the limited focusability of light beams and the formation of speckle patterns when coherent light is scattered or reflected. We use existing physical models for these phenomena to mimic the behavior of Gaussian beams and the statistical properties of speckle patterns without performing a wave-optics simulation. These models have been integrated into a physically-based renderer with the intention of synthesizing sensor data for an optical inspection setup which uses a laser light source. This requires only local changes to the renderer and can also be used for real-time rendering.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Coherent light exhibits effects caused by diffraction that cannot be described by geometrical optics. We propose a method to render a subset of diffraction effects in a physically-based renderer (see Pharr and Humphreys [PH04]) to reproduce effects observed in scenes containing coherent light. Contrary to the geometrical optics assumptions, light in general, even with a tightly limited spectrum cannot be focused onto a single point due to diffraction effects. This is especially noticeable when using laser light sources in a scene. Another well-known effect of using such light sources is the formation of speckle patterns (see Fig. 1). A speckle pattern is a granular phenomenon that occurs when coherent light is reflected or scattered by an optically rough surface; interference occurs between the waves that were scattered by different surface elements and acquired a phase difference. The emerging patterns of lower and higher intensity are called speckle patterns [Goo75]. These patterns are a characteristic and noticeable feature of reflected or scattered coherent light.

Our work is motivated by the need to synthesize sensor data for an automated optical inspection setup. Laser light sources are often used in such setups, for example in laser triangulation. Simulating the resulting sensor data is potentially very beneficial because of the complex interaction between light and material which turns setting up an inspection setup into a tedious trial and error procedure. Our motivation is to reduce the set up time by employing a physically-based renderer to simulate the light transport and thus synthetically generate sensor data. The simulated sensor data can then be used to provide insights and guidance for the set-up process or to optimize

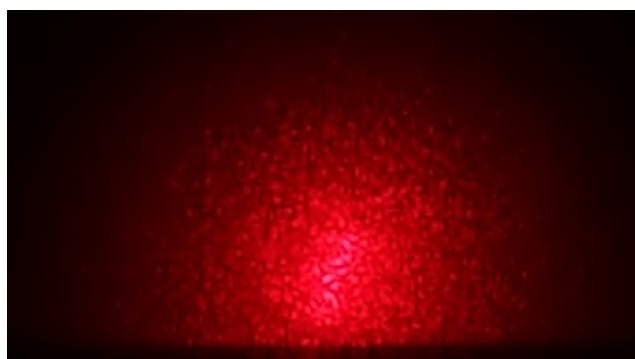


Figure 1: A photograph of a speckle pattern caused by a laser beam.

the set-up automatically. With regard to the application, it is not our intention to calculate a reference-grade solution but to mimic the mentioned properties of coherent light to a degree that can be used to qualify laser triangulation setups.

In the optical inspection setup we are working with, we use a laser emitter fitted with a lens that fans out the beam in one dimension. The laser illuminates the target object in the form of a curve and this curve is then recorded by an optical sensor. Using the captured images and based on the principle of triangulation [BLF15], a 3D measurement of the illuminated profile can be generated. One of the factors contributing to the uncertainty in the measured data are speckles [DHH94]; ignoring their effect would adversely impact the accuracy of results obtained by such a simulation. Our goal

is to include a laser as a light source in a physically-based renderer and reproduce effects of wave optics that have an impact in our scenario: the limited focusability of light beams and speckle patterns.

After reviewing related work, we will first introduce the properties of Gaussian beams and how we integrated them into the renderer (sections 3 and 4) and then continue to do the same for speckle patterns (sections 5 and 6). This is then followed by a presentation of the resulting images and a discussion of the results.

2. Related work

Our paper touches different research topics and we will present the publications most relevant to our work.

Our motivation is the simulation of an optical measurement process using computer graphic methods. The synthetization of sensor data has been a research interest for a long time and we would like to highlight some recent publications. Cajal et al. [CSS15] use computer graphic methods to simulate sensor data intended to help setting up a laser triangulation setup. They use geometrical optics and do not simulate effects of wave optics. Mohammadikaji et al. [MBI*16] use computer graphics as a tool to simulate a limited sensor resolution for uncertainty analysis in the context of laser triangulation. They account for uncertainty through wave optic effects in a summarizing manner by including such effects in an image-domain uncertainty but not explicitly modeling them. A recent publication by Retzlaff et al. [RHBD16] discusses using computer graphic methods to simulate optical sensor data and to generate synthetic sample data.

The integration of wave optics in physically-based renderers has been the subject of some works. Cuypers et al. [CHB*12] present a comparison of different approaches and propose the handling of diffraction and interference effects by introducing a new BSDF representation. Holzschuch and Pacanowski [HP15] present a reflectance model that provides a better fit to measured materials by including diffraction effects. Lindsay and Agu [LA06] render diffraction effects in real-time using spherical harmonics.

The speckle phenomenon has been studied extensively with different goals. While some publications describe ways to reduce speckles (e.g. McKechnie [McK75]) others make use of speckle patterns to deduce information about microscopic surface deformations (e.g. Ennos [Enn75] and Yamaguchi [Yam93]) or use speckle to detect touches to surfaces (Shih et al. [SDH*14]).

3. Properties of Gaussian beams

Before going into details about the integration of lasers into the physically-based renderer, we will review some necessary background information about laser beams in this section.

There are many different types of laser emitters, and the beams created by these have different properties. We limit ourselves to Gaussian beams because they are commonly used to model the behavior of laser beams. These beams are emitted by lasers working in the fundamental TEM₀₀ mode [EE10] and are preferred because they provide the best focusability.

The properties of a Gaussian beam that we want to simulate are:

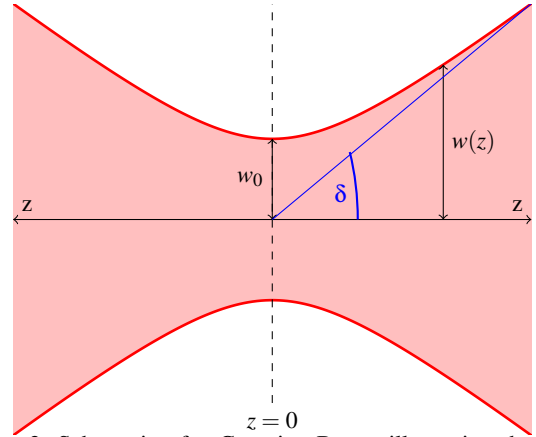


Figure 2: Schematic of a Gaussian Beam, illustrating the beam waist (where the beam width $w(z)$ is minimal) and the divergence δ .

- The width of the beam does not change linearly with traveled distance.
- Due to diffraction, a laser beam (or any other beam) cannot be focused onto a single point.

Next, we will recapitulate the governing equations for Gaussian beam propagation that are needed to model the intended behavior of the beams. The information is taken from Eichler and Eichler [EE10], please refer to this source for more detailed information.

A Gaussian beam has the defining property that its transverse intensity profile is a Gaussian. The beam width w is defined as the lateral distance from the beam center to a point where the beam intensity has decayed to $1/e^2$ of its center intensity. A Gaussian beam can be characterized solely by its wavelength λ and its beam waist w_0 . The beam waist is the width of the beam at the point where the width is minimal (the focus). The beam width is usually calculated as a function of the distance z from the beam waist, so $w_0 = w(0)$ and

$$w(z) = w_0 \sqrt{1 + \frac{z^2}{z_R^2}} \quad (1)$$

with the Rayleigh range $z_R = \frac{w_0^2 \pi}{\lambda}$ (Fig. 2). Further away from the beam waist ($z \gg z_R$), the width increases linearly with z : $w(z) \approx w_0 \frac{z}{z_R}$ with a very good approximation. Using this, the divergence angle δ can be defined as $\delta = \frac{w_0}{z_R} = \frac{\lambda}{\pi w_0}$.

For a given optical power P , we can calculate the beam center intensity with $E_0(z) = \frac{2P}{\pi w^2(z)}$. This allows us to calculate the irradiance at any point on the central axis of the beam. To determine the irradiance at off-axis points, we use the knowledge that the beam is radially symmetrical with a Gaussian profile. This leads to the expression for the irradiance at a point \mathbf{x} in space:

$$E(\mathbf{x}) = E_0(z) e^{-\frac{2r^2}{w^2(z)}} \quad (2)$$

The point $\mathbf{x} = (x, y, z)^T$ is specified with respect to the beam waist, so z is the distance from the beam waist and $r^2 = x^2 + y^2$.

In summary, completely defining a beam for rendering requires

the wavelength λ , the beam waist w_0 , the position of the beam waist and the beam direction to be given in addition to the beam's optical power P .

4. Rendering with Gaussian beam emitters

Typical physically-based renderers are based on geometrical optics and thus we cannot integrate the Gaussian beams directly like other light emitters. The usual assumption for an emitter is that the radiance of a ray can be determined if the source point and the direction are known and this is not valid in our case.

Regarding the scope of the changes that are needed, we note that when a Gaussian beam hits a surface and is reflected, its defining properties are generally lost. We do not handle lenses or mirror surfaces where these properties would be preserved, i.e. we consider only the first interaction in a coherent light path after the lens that shapes the beam (e.g. transforms the beam into a fan) in the following description. This means that there are two main cases we have to take care of: Sampling a light source directly from a point in the scene (next event estimation) and sampling a ray starting at the light source.

Before describing our approach we note that we cannot assign a radiance value to a path segment without knowing the position of both its vertices. This is not a problem in case of direct light source sampling as one vertex of the path segment is already known and the second can be chosen by our implementation before calculating the radiance. However, if we sample a point and direction on the light source, we will need to first determine the next intersection before calculating the radiance.

Next event estimation

In the context of a next event estimation from a reference point \mathbf{r} we do the following:

1. We transfer the reference point \mathbf{r} into a point \mathbf{r}_L in the local laser coordinate system where the beam is traveling from the origin in positive z direction.
2. If we are rendering a laser line, we now take into account that different types of lenses can be used to convert a laser beam into a laser fan. Cylinder lenses and Powell lenses are used in most cases with the main difference being the fact that they distribute the intensity differently along the beam. While cylinder lenses maintain the Gaussian profile along the line, Powell lenses distribute the intensity more evenly.

Our setup uses a cylinder lens and for cylinder lenses, we can just scale the local x coordinate (assuming the laser fan plane has the normal $(0, 1, 0)^T$) according to the projected length of the line at the current distance from the emitter:

$$x' = x \frac{w(z)}{\tan \frac{\alpha}{2} \cdot z}.$$

α is the opening angle of the laser line and x' is the scaled coordinate that is used for the following intensity calculation.

3. We need to determine a sampled point on the emitter aperture surface. To do this, we calculate the point on the aperture that corresponds to the current point's coordinate on the receiver plane: We calculate both the beam radii at the aperture (w_A) and

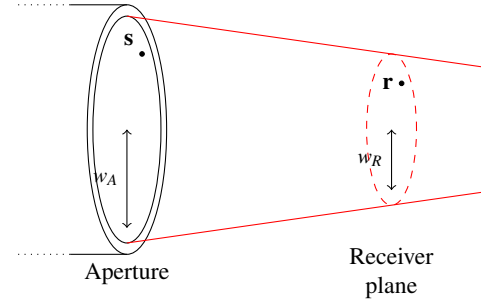


Figure 3: Determining the corresponding point on the aperture

the receiver plane (w_R) with the help of equation 1 and scale the local x and y coordinates with their ratio $\frac{w_A}{w_R}$ (see Figure 3), then translate the point along the z direction to the aperture plane. This yields a point \mathbf{s}_L in local coordinates on the aperture plane. This is then used as the sampled emitter point.

4. We are only aware of literature defining the irradiance of a laser beam, not the direction-dependent radiance. Thus, for computing the incident radiance at a surface point we assume the point \mathbf{s}_L on the aperture is the only radiance source for point \mathbf{r}_L . To determine the radiance emitted from \mathbf{s}_L to \mathbf{r}_L , we first calculate the irradiance $E(\mathbf{r})$ according to equation 2 and then calculate the radiance as

$$L(\mathbf{s}_L \rightarrow \mathbf{r}_L) = \frac{1}{\cos \theta_s} E(\mathbf{r}_L), \quad (3)$$

where θ_s is the angle between the surface normal at point \mathbf{r} and $\mathbf{s} - \mathbf{r}$. We justify the assumption of \mathbf{r} receiving only radiance from one direction by the geometrical consideration that the aperture diameter is usually in the submillimeter range while working distances of the laser are in the tens of centimeters. This means the solid angle subtended by the aperture is very small which makes this approximation feasible.

Sampling a direction from the light source

When using bidirectional methods or using light tracing we will also need to sample a position on the light source and a direction. Again, there is the restriction that we cannot determine the radiance for the path segment starting at the emitter without knowing its end.

We start by choosing a point on the emitter by uniformly sampling the disc. The beam divergence can be used to restrict the sampled directions. However, as the beam divergence angle θ describes the cone of directions which only contains beam intensities down to $1/e^2$ of the center intensity, directly using the divergence angle would result in a loss of energy, introducing a bias. This can be mitigated by calculating a modified divergence angle θ' that includes beam intensities down to a user-defined fraction $l \in (0, 1]$ of the peak intensity: $\theta' = \theta \sqrt{-0.5 \ln l}$. When sampling a direction, we can take the conservative cone of directions given by θ' and apply it to the emitter aperture (see Fig. 4). The cone is conservative in the sense that the beam usually first converges to its waist before expanding again. If rendering a laser line, the cone will be scaled along the x direction according to the laser line opening angle α . An alternative, unbiased approach would be to sample all outgoing directions, e.g. using a normal distribution resembling the beam intensity.

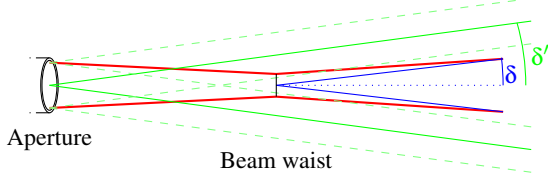


Figure 4: Limiting the sampled direction by applying the modified divergence cone at the aperture

After choosing the direction, we perform an intersection test of the ray defined by the point on the aperture and the sampled direction with the scene geometry. If there is an intersection, we can determine the radiance according to equation 3 and therefore meet the requirement of a typical physically-based renderer.

The described approaches require only a custom emitter to be included in most physically-based rendering frameworks. Before we demonstrate some images rendered with these changes in section 7 we will first cover the second main topic, simulating speckle patterns.

5. Formation of speckle patterns

As with the Gaussian beams, before going into the rendering details, we will here first summarize the properties of speckle patterns most important for rendering. The information was taken from Goodman [Goo75] and we refer to this in-depth treatment of the statistical properties of speckle pattern for more details.

Speckle patterns form when coherent light such as laser light is reflected by an optically rough surface. Optically rough means that the surface is rough in wavelength scale and can be assumed to cause phase differences uniformly distributed in the interval $[0, 2\pi)$. Under these circumstances, light that arrives in-phase at the surface is reflected by individual surface elements and the different distances traveled by the light due to the spatial differences of the surface elements cause relative delays between the different waves. These delays then cause a spatial pattern of constructive or destructive interference which can be observed as brighter or darker areas in the reflected light. The emerging patterns are called speckle patterns (Fig. 1). Goodman shows that the statistics of the speckle pattern do not depend on the actual surface material as long as the surface is optically rough, a fact that allows us to create the speckle patterns independently of the surface materials.

Speckle patterns can be categorized into two different types according to their formation: Objective (Fig. 5a) and subjective speckle patterns (Fig. 5b).

Objective speckle patterns are formed when only free-space propagation of the wavefronts is considered. They can be observed e.g. by placing a scattering screen in the path of the reflected light.

Subjective speckle patterns are formed when an imaging system is used to observe the speckle pattern directly (without intervening screen). Here a lens with an aperture is placed between the reflecting surface and the plane (e.g. sensor) on which the speckles are observed.

6. Rendering speckle patterns

6.1. Problem definition

Our goal is to render speckle patterns in a measurement simulation context. However, we have no knowledge of the micro-geometry of the surface and thus cannot recreate the exact speckle patterns that would be created by the coherent light. Instead, our goal is that the simulated patterns exhibit statistical properties and behavior as predicted by speckle theory, described by Goodman [Goo75]. In the following, we make the same assumptions as Goodman:

1. The light is perfectly coherent and polarized.
2. The reflecting surface is optically rough. This assumption is valid for a wide range of surfaces, because most surface structures are large compared to the wavelength of light ($\approx 700 \times 10^{-9}$ m). Only special manufacturing techniques create smoother surfaces (e.g. honing).
3. The strength of a reflection is unrelated to its phase.
4. The surface does not depolarize the light when reflecting it.
5. In his derivations Goodman also uses the Fresnel approximations which are valid for the near field, i.e. points near the aperture.

Under these assumptions, speckle patterns exhibit the following characteristic properties, which we include in our simulation:

Intensity distribution As Goodman [Goo75] shows, the intensity distribution of a speckle pattern follows negative exponential statistics. The PDF of the intensity has the form

$$p(I) = \begin{cases} 1/\bar{I} e^{-I/\bar{I}} & I \gg 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where \bar{I} is the mean intensity.

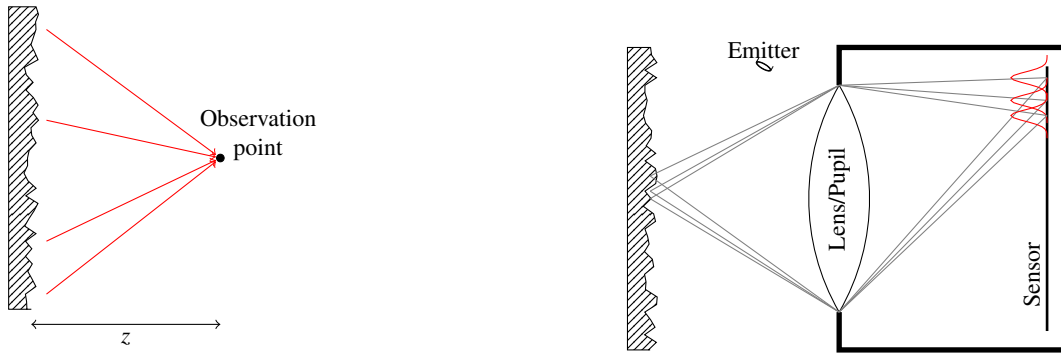
Frequency distribution The frequency distribution of a speckle pattern is dependent on the set of interfering waves and changes with the shape and size of the illuminated patch on the surface.

Speckle size It can be shown that the size of the smallest spots in the granular speckle pattern is determined by the wavelength of the illumination and the geometric configuration of the reflecting surface and the observer (see [Goo75] for details).

Gradual decorrelation The speckle pattern changes gradually as the elements participating in its formation (like the surface or the camera) change their relative positions.

We are (virtually) observing the speckle patterns with an imaging system and thus we need to consider only subjective speckle patterns. The statistics of such a speckle pattern depend only on the camera geometry [Goo75] because we can essentially consider the pupil area as the source of many individual waves that have a phase difference uniformly distributed in $[0, 2\pi)$. This assumption is valid because the (objective) speckles forming on the pupil plane are small when compared with the diameter of the pupil. The propagation from pupil to sensor can then be handled as a free space propagation. Therefore the pattern in this case does not depend on the illuminated surface area but the pupil area and the distance between the pupil and the observation plane/sensor. Since these parameters are usually fixed before the rendering starts, we can generate the pattern in a preprocessing step.

The general flow of our approach is then to first create the two-dimensional pattern (section 6.2) as a preprocessing step before the



(a) Objective speckle formation: The different phases of the reflected light caused by surface height variations lead to constructive or destruction interference at the observation point.

(b) Subjective speckle formation: The de-phased contributions of many point-spread functions on each point of the sensor cause interference

Figure 5: Speckle formation geometry.

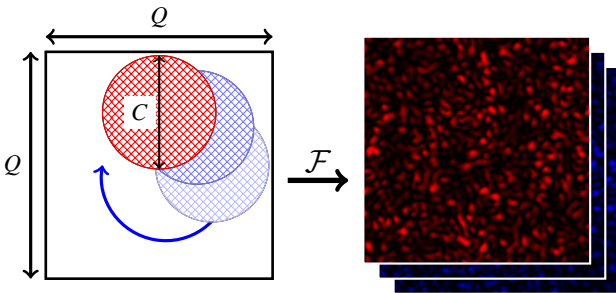


Figure 6: Two-dimensional array of the size Q with embedded circular random field with diameter C for speckle pattern generation (red) and with shifted random fields for decorrelated patterns (blue).

rendering starts. During the rendering, the pattern is used as a look-up table. We generate a look-up coordinate to access the pattern and use the stored value to modulate the pixel contribution. We describe two variants here: The basic variant (section 6.3) describes the application of the pattern without taking the gradual change of the pattern into account. Based on the basic application, section 6.4 then describes how gradual pattern decorrelation can be integrated into the system.

6.2. Pattern generation

Duncan and Kirkpatrick [DK08] proposed a technique to generate speckle patterns that we will employ here: In a zeroed two-dimensional array of $Q \times Q$ pixels, we generate a circular field of random complex numbers with a magnitude of one (see Fig. 6). The phase (imaginary part) must be uniformly and independently distributed in $[0, 2\pi)$. Then we apply a Fourier transformation to the complete array. Taking the element-wise squared magnitude of the complex result array yields the two-dimensional scalar-valued speckle pattern. This array is then scaled so that its entries can be directly used as modulators conforming to equation 4.

The ratio $\frac{Q}{C} = h_p$ of the size Q of the square array to the diameter C of the random field (Fig. 6) determines the size of the smallest speckle in pixels in the generated pattern. It follows from Goodman's work [Goo75] that the minimum speckle size is $h = \frac{\lambda d_{PI}}{D}$

where λ is the wavelength of the coherent light, d_{PI} is the distance between the pupil and the sensor and D is the diameter of the pupil. This means that the smallest speckle extends over $h_I = \frac{h}{w_{\text{pixel}}}$ sensor pixels if w_{pixel} is the pixel size of the sensor. C and Q can be chosen by the user with the restriction that the largest circle valid diameter is $\frac{Q}{2}$. We will later use this correspondence between the smallest speckle h_I and the smallest speckle h_p in the pattern to determine the coordinate for the pattern lookup.

6.3. Basic Pattern application

The precomputed pattern is applied during rendering. Whenever a contribution of coherent light is encountered, that contribution is multiplied with a value from the pattern. To access the pattern, we use a pattern coordinate \mathbf{c} that is two-dimensional in the basic case. If a component of the pattern coordinate \mathbf{c} falls outside the range $[0, Q - 1]$ the coordinate is wrapped around to tile the pattern (as per construction the generated pattern is seamlessly tileable). We use bilinear interpolation to determine the pattern value for non-integral coordinates.

The calculation of \mathbf{c} is motivated by the following considerations: Sjö Dahl [Sjö95] derives the displacement of the speckles on the sensor \mathbf{a}_I from the displacement \mathbf{a} on the object for the general case. We do not consider out-of-focus speckles, thus we use a simplified version of his expression:

$$\mathbf{a}_I = -\frac{d_{PI}}{d_{SP}} \begin{pmatrix} \cos q_1 & 0 & \sin q_1 \\ \sin q_1 \sin q_2 & \cos q_2 & \cos q_1 \sin q_2 \end{pmatrix} \mathbf{a}. \quad (5)$$

Here, q_1 and q_2 are angles that describe the transformation (neglecting any translation) between the two Cartesian surface coordinate systems \mathbf{n} and \mathbf{b} (Figure 7). \mathbf{n} is the surface coordinate system with \mathbf{n}_3 being the surface normal and $\mathbf{n}_1, \mathbf{n}_2$ lying on the surface plane. \mathbf{b} is analog for the sensor plane. q_1 describes the rotation around \mathbf{n}_2 while q_2 describes the following rotation around the first axis of the intermediate system (Euler angles). Note that like other authors in this field, Sjö Dahl uses a series of approximations (including the paraxial approximations) to derive the speckle displacements. When describing imaging geometry, he assumes a single-lens system. In this work, we follow this assumption. Additionally when using results from Sjö Dahl or Li and Chiang [LC86]

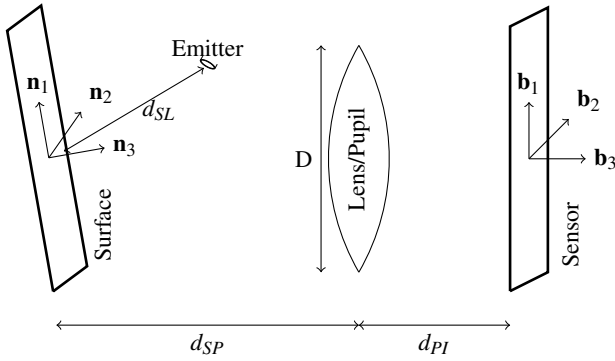


Figure 7: Geometric definitions for subjective speckle.

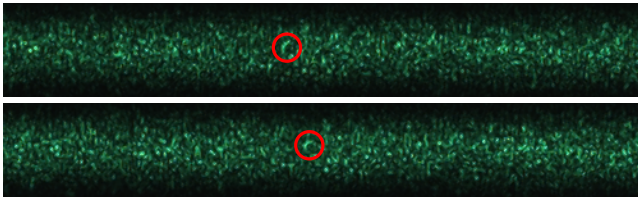


Figure 8: Speckle displacement and decorrelation. Both images are photographs of speckles produced by a laser line. In the lower one, the surface has been moved by 0.25mm. Note how the marked shape has been translated and its form has changed as an example of pattern displacement and decorrelation.

in this or the next section, we assume an aberration-free lens. For details we refer to the original publications.

We calculate the displacement of the object in the camera's coordinate system according to equation 5, then use the obtained offset in addition to the two-dimensional screen-space position \mathbf{t} of the pixel where the pattern should be applied to access the pattern. We then access the speckle pattern coordinate $\frac{h_P}{h_I} \mathbf{a}_I + \frac{h_P}{h_I} \mathbf{t}$ where h_I is the minimum speckle size in sensor pixels and h_P is the minimum size in the generated pattern as described above. This will result in a speckle pattern that is stable in screen space and has a speckle size that matches the camera geometry. Note that due to the two-dimensional nature of our generated pattern, only motion in the plane of the camera results in realistic pattern movement while other transformations will produce a displacement of the pattern that is in general not physically correct. However, the decorrelation behavior described in the next section takes more possible transformations into account.

6.4. Pattern correlation

When the camera (or other elements participating in the formation of the speckle pattern) change their spatial relation, the pattern also moves but changes gradually at the same time until there is no correlation between the new and the old pattern anymore (see Figure 8). This is the behavior we intend to mimic in our system. However, generating a speckle pattern is a computationally intensive process and again we use preprocessing to avoid the cost involved with creating a pattern from scratch.

During the preprocessing, instead of creating a single pattern like described in section 6.3, we create a stack of $2M$ speckle patterns

(referred to as slices) with progressively less correlation. Duncan and Kirkpatrick [DK08] describe that patterns with a varying degree of correlation can be generated by treating the circular patch we used when creating the pattern as a circular mask into a random field and translating that mask. The degree of correlating between two generated patterns depends on the amount of overlap between the two masks. For each of the patterns we translate the circular mask, creating a series of $2M$ patterns. We create the patterns such that they have a cyclic correlation, but the central pattern in the series is completely decorrelated from the first, or in general, pattern i is completely decorrelated from $i + M$. This is done by moving the circular mask on a circular path through the field of random numbers (see the blue masks in Figure 6). This means that the later patterns will become increasingly correlated with the first again. This is done to guarantee a smooth change of the pattern when the coordinate is wrapped around (see discussion in section 7).

The coordinate \mathbf{c} to access the pattern is now extended by a third component u used to access the slices. In case of non-integral values of u we use linear blending between adjacent slices to determine the final pattern value. Like the first two components we wrap u around if it falls outside the $[0, 2M - 1]$ interval.

When encountering a coherent light contribution to a pixel, we again have to determine the pattern coordinate \mathbf{c} . The first two components of \mathbf{c} are determined like described in section 6.3. The third coordinate signifies the degree of correlation between the speckle patterns, thus we need a way to quantify this.

Yamaguchi [Yam85] introduced the correlation factor $\gamma_0 \in [0, 1]$ to quantify the correlation between speckle patterns. We will use the correlation factor to determine the distance in the u coordinate that two different configurations of camera and surface should have from each other. Calculating the speckle pattern correlation factor for all possible changes to the light path (e.g. translation, rotation and general surface deformation) is a complex task and out of the scope of our work. We will concentrate on modeling three causes for decorrelation because they are most common in our scenario: in-plane camera movement, surface movement and distance change. Of those, the first two must be handled in combination, because they can counter each other. Both cause a displacement of the speckle pattern in the aperture plane and according to Sjö-dahl [Sjö95] this is a measure of the decorrelation as the decorrelation factor is the square of the normalized overlapping area of two discs (representing the aperture). Thus we calculate a displacement of the speckle pattern in the aperture for these two cases and a direct correlation coordinate contribution for the distance change.

In-plane camera movement When the camera is moved in the lens plane, its aperture moves like a mask through the objective speckle field in space [SDH*14]. When the apertures no longer overlap, the pattern is uncorrelated to that at the start of the translation. The amount of in-plane movement is represented by a vector \mathbf{a}_P^C .

Surface movement Sjö-dahl [Sjö95] also derives the displacement \mathbf{a}_P^S that is caused by an object displacement. He gives the two

components of the aperture displacement as

$$\begin{aligned} \mathbf{a}_{P,x}^S &= \frac{-d_{SP}}{\cos(q_1)} \left[a_x \left(\frac{1_{Sx}^2 - 1}{d_{SL}} + \frac{1_x^2 - 1}{d_{SP}} \right) + \right. \\ & a_y \left(\frac{1_{Sx}1_{Sy}}{d_{SL}} + \frac{1_x1_y}{d_{SP}} \right) + a_z \left(\frac{1_{Sx}1_{S_z}}{d_{SL}} + \frac{1_x1_z}{d_{SP}} \right) \left. \right] + \\ & d_{SP} \frac{\sin(q_1)\sin(q_2)}{\cos(q_1)\cos(q_2)} \left[a_x \left(\frac{1_{Sx}1_{Sy}}{d_{SL}} + \frac{1_x1_y}{d_{SP}} \right) + \right. \\ & a_y \left(\frac{1_{Sy}^2 - 1}{d_{SL}} + \frac{1_y^2 - 1}{d_{SP}} \right) + a_z \left(\frac{1_{Sy}1_{S_z}}{d_{SL}} + \frac{1_y1_z}{d_{SP}} \right) \left. \right] \end{aligned}$$

and

$$\begin{aligned} \mathbf{a}_{P,y}^S &= \frac{-d_{SP}}{\cos(q_2)} \left[a_x \left(\frac{1_{Sx}1_{Sy}}{d_{SL}} + \frac{1_x1_y}{d_{SP}} \right) + \right. \\ & a_y \left(\frac{1_{Sy}^2 - 1}{d_{SL}} + \frac{1_y^2 - 1}{d_{SP}} \right) + a_z \left(\frac{1_{Sy}1_{S_z}}{d_{SL}} + \frac{1_y1_z}{d_{SP}} \right) \left. \right]. \end{aligned}$$

The symbols $1_{Sx}, 1_{Sy}, 1_{S_z}$ and $1_x, 1_y, 1_z$ correspond to the components of the normalized vector from the surface to the laser emitter and of \mathbf{b}_3 , respectively. Both are expressed in terms of the coordinate system $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$.

Distance change Li and Chiang [LC86] have calculated correlation factors for different types of changes to the geometrical configuration. For a changing distance between the surface and the camera (assuming a circular aperture) they give the correlation factor as

$$\gamma_0 = \left| \frac{\sin(\frac{\pi}{2} \mathbf{a}_O)}{\frac{\pi}{2} \mathbf{a}_O} \right|^2,$$

with

$$\mathbf{a}_O = \frac{1}{\left[2 \left(1 + \frac{1}{m} \right) F \right]^2} \frac{d_z}{\lambda}$$

being the out-of-plane pattern displacement of the speckles, d_z the displacement along the optical axis, m the magnification factor and F the F-Number of the aperture. It follows that for $|d_z| = \lambda \left[2 \left(1 + \frac{1}{m} \right) F \right]^2$ the pattern is decorrelated. This distance will be referred to as d_z^0 .

We can now determine the u coordinate by calculating the effective speckle displacement in the aperture plane and combining this with the decorrelation caused by the distance change:

$$u = \left(\frac{|\mathbf{a}_P^C + \mathbf{a}_P^S|}{D} + \frac{d_z}{d_z^0} \right) M.$$

Using the completed pattern coordinate we access the stack of speckle pattern to retrieve the correct interference factor and modulate the coherent light contribution to the pixel.

7. Results and discussion

We integrated our techniques into Mitsuba [Jak10], a research-oriented physically-based renderer, which we then used to generate all the results in this paper. All images have been rendered using 256 samples per pixel. Integrating the Gaussian Beam rendering only required the addition of an emitter plugin in Mitsuba without

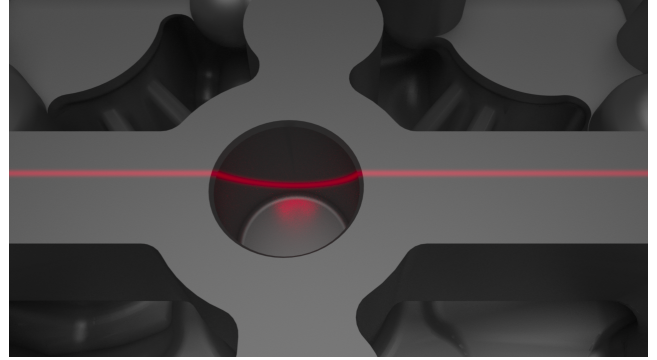


Figure 9: A rendered laser line reflected by a metallic surface.

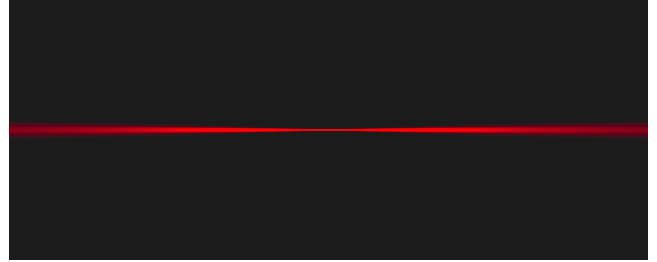


Figure 10: Rendered detail image of the beam waist of a beam that is nearly parallel to a Lambertian surface.

modifying the core of the renderer. For speckles we added a flag to the core to differentiate between coherent and non-coherent emitters. Other than that, speckles have been realized by sub-classing and extending an existing integrator plugin. The new integrator pre-calculates the patterns and applies them before the radiance values are written into the image storage.

With the changes in the physically-based renderer we described, we can render the effects of beams of coherent light reflected on surfaces and take into account the desired effects of physical optics. A laser line reflected on a metallic surface is shown in Figure 9. In Figure 10 the parameters of the laser emitter have been selected to produce a clearly noticeable beam waist. Note that speckles have not been enabled in these two figures. Figure 11 demonstrates the effects of enabling speckle rendering. The parameters here have been selected to make the speckles clearly visible while in Figure 12 the effect is more subtle.

Approximation errors We approximate the radiance reaching a point within the beam by assuming that all the radiance arrives from a single direction. This will cause errors if the emitter radius is very large or the emitter is very close to the surface. However, in most realistic scenarios (for example in our measurement setup), this is a reasonable constraint and the error introduced by this assumption is not significant. In cases where the constraints are not satisfied, the incoming radiance will be spread over a larger solid angle and not just the single direction we assumed. This will cause the reflected radiance to differ from our calculations. However, the exact manifestation of the error will depend on the surface BRDF and the viewing angle so the error will have to be evaluated for a given situation.

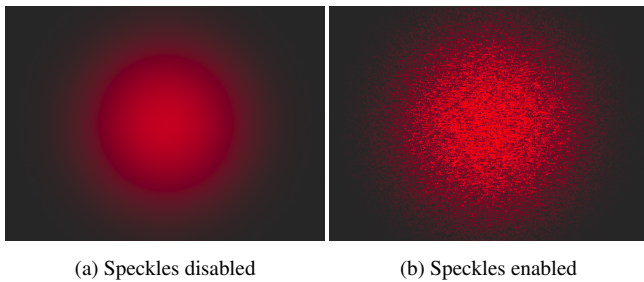


Figure 11: Rendered images of laser spots reflected on a Lambertian surface, rendered with and without speckles but otherwise using the same settings.



Figure 12: Rendered laser line with speckle pattern.

Pattern size Care must be taken to choose the pattern size Q large enough so that no disturbing repetition artifacts can be seen. The appropriate size depends on the application and geometry parameters.

Correlation Similar to the pattern size, the cyclic nature of the decorrelated slices means that patterns start becoming correlated again when the u coordinate increases. This effect can be mitigated by making sure the cycle is long enough to satisfy the need of the application. There is a trade-off here between longer preprocessing time and increased memory consumption and correlation faithfulness.

8. Conclusions

We have shown how we recreate certain effects like diffraction-limited focusing and speckle formation within the framework of a physically-based renderer based on the geometrical optics assumption. While these effects could be simulated on a more detailed level, doing so will require substantially more computational power and a phenomenological simulation like demonstrated in the paper might be more feasible depending on the intended use. Moreover, the demonstrated techniques can also be integrated in a real-time rendering system without requiring substantial changes.

9. Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. This work has been funded by DFG grant DA 1200/3-1.

References

- [BLF15] BEYERER J., LEÓN F. P., FRESE C.: *Machine Vision: Automated Visual Inspection: Theory, Practice and Applications*. Springer, 2015. 1
- [CHB*12] CUYPERS T., HABER T., BEKAERT P., OH S. B., RASKAR R.: Reflectance model for diffraction. *ACM Transactions on Graphics* 31, 5 (2012), 1–11. 2
- [CSS15] CAJAL C., SANTOLARIA J., SAMPER D.: Simulation of Laser Triangulation Sensors Scanning for Design and Evaluation Purposes. *International Journal of Simulation Modelling* 14, 2 (2015), 250–264. 2
- [DHH94] DORSCH R. G., HÄUSLER G., HERRMANN J. M.: Laser triangulation: fundamental uncertainty in distance measurement. *Appl. Opt.* 33, 7 (Mar 1994), 1306–1314. 1
- [DK08] DUNCAN D., KIRKPATRICK S.: Algorithms for simulation of speckle (laser and otherwise). *Proc. SPIE* 6855, January (2008), 685505–1–685505–8. 5, 6
- [EE10] EICHLER J., EICHLER H. J.: *Laser - Bauformen, Strahlführung, Anwendungen*. 2010. 2
- [Enn75] ENNOS A.: Speckle Interferometry. In *Laser speckle and Related Phenomena*, vol. 9. 1975, ch. 6, pp. 203–254. 2
- [Goo75] GOODMAN J. W.: Statistical Properties of Laser Speckle Patterns. In *Laser speckle and Related Phenomena*, vol. 9. 1975, ch. 2, pp. 9–75. 1, 4, 5
- [HP15] HOLZSCHUCH N., PACANOWSKI R.: *A physically accurate reflectance model combining reflection and diffraction*. Research Report RR-8807, INRIA, Nov. 2015. 2
- [Jak10] JAKOB W.: Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>. 7
- [LA06] LINDSAY C., AGU E.: Physically-based real-time diffraction using spherical harmonics. In *Proc. of the Second International Conference on Advances in Visual Computing - Volume Part I* (2006), ISVC'06, Springer-Verlag, pp. 505–517. 2
- [LC86] LI D. W., CHIANG F. P.: Decorrelation Functions in Laser Speckle Photography. *Journal of the Optical Society of America A-Optics Image Science and Vision* 3, 7 (1986), 1023–1031. 5, 7
- [MBI*16] MOHAMMADIKAJI M., BERGMANN S., IRGENFRIED S., BEYERER J., DACHSBACHER C., WÖRN H.: A Framework for Uncertainty Propagation in 3D Shape Measurement using Laser Triangulation. *2016 IEEE International Instrumentation and Measurement Technology Conference* (2016), 6–11. 2
- [McK75] MCKECHNIE T. S.: Speckle Reduction. In *Laser speckle and Related Phenomena*, vol. 9. 1975, ch. 4, pp. 123–169. 2
- [PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory To Implementation*. Morgan Kaufmann series in interactive 3D technology. Elsevier Science, 2004. 1
- [RHBD16] RETZLAFF M.-G., HANIKA J., BEYERER J., DACHSBACHER C.: Potential and Challenges of Using Computer Graphics for the Simulator of Optical Measurement Systems. *18. GMA/ITG Fachtagung: Sensoren und Messsysteme 2016* (2016), 322–329. 2
- [SDH*14] SHIH Y., DAVIS M. A., HASINOFF S. W., DURAND F., FREEMAN W. T.: Laser Speckle Photography for Surface Tampering Detection. *Proc. 25th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012 1* (2014). 2, 6
- [Sjö95] SJÖDAHL M.: Calculation of speckle displacement, decorrelation, and object-point location in imaging systems. *Applied optics* 34 (1995), 7998–8010. 5, 6
- [Yam85] YAMAGUCHI I.: Fringe formations in deformation and vibration measurements using laser light. vol. 22 of *Progress in Optics*. Elsevier, 1985, pp. 271 – 340. 6
- [Yam93] YAMAGUCHI I.: Theory And Applications of Speckle Displacement and Decorrelation. In *Speckle Metrology*. CRC Press, 1993, pp. 1–40. 2